John von Neumann Institute for Computing

Subspace methods for sparse eigenvalue problems

Bernhard Steffen

published in

Modern Methods and Algorithms of Quantum Chemistry,
J. Grotendorst (Ed.), John von Neumann Institute for Computing, Jülich, NIC Series, Vol. 1, ISBN 3-00-005618-1, pp. 279-286, 2000.

http://www.fz-juelich.de/nic-series/

SUBSPACE METHODS FOR SPARSE EIGENVALUE PROBLEMS

B. STEFFEN
John von Neumann Institute for Computing
Central Institute for Applied Mathematics
52425 Jülich
Germany
E-mail: b.steffen@fz-juelich.de

Subspace methods are the methods of choice for calculating a few eigenvalues and -vectors of a large matrix. They may also be considered for completely diagonalizing a matrix if it is either sparse or too large to be stored. A subspace method for a $n \times n$ matrix A consists of a scheme to extract approximations to some eigenvalues and -vectors of A from the action of A onto a subspace $\mathbf{V} \subset \mathbb{C}^{\mathbf{n}}$ and a method to update $\mathbf{V}$. We will present extraction schemes for extremal (Ritz projection) as well as inner (residual minimization, harmonic Ritz projection) eigenvalues and discuss advanced update schemes.

1 Introduction

The problem of calculating some or all eigenvalues and -vectors of a large matrix appears in a wide range of applications from biology (meta stable states of ecosystems) to mechanical engineering (oscillations of suspension bridges). Theoretical chemistry methods have an outstanding position here by giving rise to extremely large matrices with very special properties, so that there are methods that are very successful there but rarely in use - and not very effective - with problems from other sources. As eigenvalue problems, and particularly eigenvalue problems from theoretical chemistry, make up a fair share of the supercomputer usage, there is good reason for analyzing and improving the algorithms and implementations as well as for teaching users to make the most of the method chosen.

The classification of eigenvalue problems distinguishes between different matrix structures (general, hermitian, complex symmetric, sparse, ...), different requirements (eigenvalues only, eigenvectors too, all eigenvalues, only extremal eigenvalues, certain part of spectrum, eigenvector similar to excitation vector, ...) and numerical properties (normal, diagonally dominant, ...). While these are extremely important for issues of efficiency and implementation, there are only three basic principles involved in the solution of eigenvalue problems:

Similarity transformations $A = S^{-1}JS$ converting A into some normal form (usually diagonal or Jordan) where eigenvalues can be extracted immediately and eigenvectors are given by columns of S. S is built up iteratively as a product of simple matrices.

Subspace methods where A is projected onto a low (e.g. m) dimensional subspace $\mathbf{V}$. If $\hat{A}$ is the orthogonal projection of A onto $\mathbf{V}$ ($\hat{A} = W^T AW$, where W is an othonormal basis of $\mathbf{V}$), the eigenvalues of $\hat{A}$ give approximations to those of A and its eigenvectors t_i multiplied by W can be used as approximations $\bar{x}_i$ to eigenvectors of A (see[9]). For hermitian A, these approximations are almost optimal for the extremal eigenvalues of A (Ritz projection), for non hermitian A, they are still

about the best thing available. Approximations λ_i to the inner eigenvalues near a value $\bar{\lambda}$ can be constructed either from the inverse of the orthogonal projection of $(A - \bar{\lambda}I)$ onto $\mathbf{V}$ (harmonic Ritz projection) or by minimizing some function $f((A - \lambda I)v, \lambda - \bar{\lambda})$ over $v \in \mathbf{V}$ and $\lambda \in \mathbb{C}$ (residual minimization). With the information from the $\bar{x_i}$ and $(A - \bar{\lambda_i}I)\bar{x_i}$, a new (possibly larger and hopefully better) subspace $\mathbf{V}'$ can be constructed.

Nonlinear equation methods treat the problem directly as an $(n+1)$-dimensional nonlinear equation, searching for a solution in the neighborhood of $\bar{\lambda}$. Methods in use are 'shift and invert' and polynomial iteration $x^n = P_n(A)x^{n-1}$, where the polynomials P_n are chosen such that the sequence x^n converges, e.g. to the eigenvector of the largest eigenvalue.

The subspace methods differ primarily in the update procedure. New directions are added to $\mathbf{V}$, old directions may be removed. The new directions may be random (not recommended), $A\mathbf{V}$, (Lanczos, Arnoldi[4,6]), $(A - \bar{\lambda}I)^{-1}\bar{x_i}$, or various approximations to $(A - \bar{\lambda}I)^{-1}((A - \lambda_i I)\bar{x_i})$[5,7,12,3]. The latter are the most interesting methods and - notably for theoretical chemistry problems - give the best performance.

All subspace methods separate the computations into a low dimensional nonlinear problem and a number of linear algebra operations (matrix × vector and scalar products) in n dimensions. The latter make up only a small part of the code, but take most of the time. The implementation has to be carefully tuned to reduce the number of operations in n dimensions, low dimensional computations being of minor importance. The linear algebra operations should be carefully optimized (e.g. using BLAS 3), and they allow a large amount of parallelization, too. Parallelizing the low dimensional part is in the process of development, but not really useful yet[3].

2 Eigenvalue extraction

The goal of the extraction part is to find in a given subspace vectors that are good approximations of eigenvectors of A as well as the corresponding approximations of the eigenvalues. The description is independent of the update method, while the implementation shows some interdependence.

2.1 Ritz projection

The Ritz projection is the most important approach to extract eigenvalue and -vector approximations from a given subspace. The basic idea[10] is: May $\mathbf{V^{(k)}}$ be a subspace of $\mathbb{R}^n$ at iteration step k with an orthonormal basis $\vec{w}_1^{(k)}, \ldots, \vec{w}_m^{(k)}$ and $W^{(k)}$ the matrix with columns $\vec{w}_j^{(k)}$, $S^{(k)} := (W^{(k)})^T A W^{(k)}$, $\bar{\lambda}_j^{(k)}$ the eigenvalues of $S^{(k)}$, and $T^{(k)}$ a matrix with the eigenvectors of $S^{(k)}$ as columns. The columns $\vec{x}_j^{(k)}$ of $W^{(k)} T^{(k)}$ (the Ritz vectors) are approximations to eigenvectors of A with the Ritz values $\bar{\lambda}_j^{(k)} = (\vec{x}_j^{(k)})^T A \vec{x}_j^{(k)}$ approximating eigenvalues of A. If the subspace allows a good approximation of the extremal eigenvectors of A, the corresponding Ritz vectors will be close to optimal approximations[10].

For hermitian A, the Ritz vectors are forced to be orthogonal, while the projections of the eigenvectors of A onto $\mathbf{V^{(k)}}$ will not be orthogonal. Now, the Ritz vector $\vec{x}_1^{(k)}$ to the smallest (largest) Ritz value may be askew to all eigenvectors of A. $\vec{x}_2^{(k)}$ will be orthogonal to $\vec{x}_1^{(k)}$, therefore even if a non extremal eigenvector of A has a good approximation in $\mathbf{V^{(k)}}$, this may not be orthogonal to $\vec{x}_1^{(k)}$ and therefore not be close to a Ritz vector. Therefore, the second eigenvector of A can be expected to be well approximated only if the extremal one has at least a decent approximation, and inner eigenvectors of A may be poorly approximated even if a good approximation is contained in $\mathbf{V^{(k)}}$. This effect is pronounced when there appear numerically multiple Ritz values.

Example: Let A be a diagonal matrix and let $\mathbf{V^{(k)}}$ contain a good approximation to an inner eigenvector of A:

$$A := \begin{bmatrix} -1000 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix} \qquad \mathbf{V^{(k)}} := \mathit{span}\ \left(\begin{bmatrix} 0.001 & -0.005 \\ 0.1 & -1.0 \\ 1 & 0.001 \\ 0 & 1.0 \end{bmatrix} \right)$$

The approximations to $[0,0,1,0]$ calculated by Ritz projection will be $[-.00184, -.4423, .7338, .5156]$, which is almost 43^0 off the desired eigenvector and much inferior to the starting approximation. The Ritz value of 4.92 is almost correct. The situation is not necessarily improved by improving the subspace. Changing the first column of $\mathbf{V^{(k)}}$ to $[0.0001, 0.01, 1, 0]$ does not help.

2.2 Methods for interior eigenvalues

If Ritz projection performs poorly, inner eigenvalues may be approximated from a subspace containing a good approximation of the eigenvector by either of two methods depending on information available. Instead of calculating the projection of A onto $\mathbf{V^{(k)}}$, an inverse projection of $(A - \bar{\lambda}I)^{-1}$ onto $\mathbf{W^{(k)}} := (A - \bar{\lambda}I)\mathbf{V^{(k)}}$ is calculated with only marginally increased effort. Now the formerly interior eigenvalues transform to extremal ones, and if $\bar{\lambda}$ is chosen properly, the corresponding eigenvector approximations (in $\mathbf{W^{(k)}}$) are good. Applying $(A - \bar{\lambda}I)^{-1}$ to these approximations is easy, just a linear combination of the basis vectors in $\mathbf{V^{(k)}}$, and yields good approximations to eigenvectors in the neighborhood of $\bar{\lambda}$ (harmonic Ritz projection, [12]). With $\bar{\lambda} = 4.9$, the previous example returns $[0.0017, 0.2390, .9605, -0.1429]$, a much better but not optimal approximation. Changing $\bar{\lambda}$ to 4.995 gives $[0.0013, 0.1630, 0.9845, -0.0646]$, quite good. This will work well when eigenvalues in a well-known range are looked for.

There are, however, problems where the eigenvalues are not known with sufficient accuracy, but approximations to the eigenvectors are known, e.g. from low accuracy computations or from observation. In this case, a residual minimization gives better results:

Let $\bar{v} \in \mathbf{V^{(k)}}$ be an approximation to an eigenvector $\bar{x}$ with $||\bar{v}|| = 1$.

Choose $x^{(k)}$ as the local minimum of $||(A - x^T Ax \;\; I)x||$ for all $x \in \mathbf{V^{(k)}}$, $x^T v/(||x||||\bar{v}||) > \alpha$, that is closest to $\bar{v}$.

Usually, $\alpha > 0.9$ is safe, $\sqrt{0.5}$ being the theoretical lower limit. This means looking for an eigenvector that is closer to $\bar{v}$ than any other eigenvector. In the non-hermitian case, the threshold may have to be increased depending on the angle between eigenvectors of A. As this is a (low dimensional) nonlinear problem, some approximation is needed. A simple but almost always sufficient linearization is minimizing $||(A - \bar{v}^T A\bar{v}\ I)(\bar{v} + x)||$ over $x^T\bar{v} = 0$. This yields $[0.00006, -0.0861, .9791, .1840]^T$ for the above example, only slightly less accurate than harmonic Ritz projection with $\bar{\lambda} = 4.995$. Changing the $V[1,2]$ to 0.0001 changes the picture, the harmonic Ritz projection will perform only slightly better than simple Ritz projection, while the residuum minimization will be near perfect. In general, residuum minimization is more robust than harmonic Ritz projection and therefore may be a good choice for starting steps, but it is in most cases tested inferior in final convergence.

3 Update procedures

The update procedure creates $\mathbf{V^{(k+1)}}$ from $\mathbf{V^{(k)}}$ by adding some directions and possibly reducing the dimensions again. The reduction of dimensions is increasing the number of iteration steps, but as the steps get computationally cheaper, there will usually be some gain in computing time. Even if not, the reduction in memory requirement may be helpful. The reduction - sometimes termed restart - usually retains the approximations to the eigenvectors required plus those to neighboring eigenvectors.

3.1 Krylow space updates of subspace

The simplest sequence of subspaces $\mathbf{V^{(k)}}$ is given by the Krylow construction $\mathbf{V^{(k)}} = span(x_0, Ax_0, \ldots, A^{(k)}x_0)$. This is the basis of the methods of Lanczos and Arnoldi, which are about the best possible for black box solvers for a few extreme eigenvalues. The Krylow space allows a construction of an orthogonal basis via a three term recurrence which is extremely efficient. There are quite a number of computational shortcuts available with this choice of updates, such that the performance is better than an iteration count would suggest. If only eigenvalues are required, they need little storage, while the computation of eigenvectors is either very memory-consuming or needs a second pass. There are stability problems, but those can be dealt with nicely, and efficient implementations are available [4,6,1].
A related choice is $\mathbf{V^{(k+1)}} = span(\ Aw_1{}^{(k)}, \ldots, Aw_m{}^{(k)})$. This has the advantage of a search subspace with constant dimension which reduces memory requirement and enhances stability, but converges only to the largest eigenvalues[10]. Improvements use Chebychev acceleration $\mathbf{V^{(k+1)}} = span(\ P(A)w_1{}^{(k)}, \ldots, Aw_m{}^{(k)})$ [10,13,11].

3.2 Approximate inverse updates of subspace

Approximate inverse updates use special features of the matrix and can therefore be very efficient if properly implemented. The idea is to define a linearized

correction equation of the eigenvalue approximation and utilize a computationally cheap approximation of this equation. With $\lambda_1 = x_i^T A x_i$ and e_i the correct eigenvector, this equation reads $(A - \lambda_1 I)(e_i + q_i) \approx (A - \lambda_1 I)x_i$, which with $\lambda_1 \approx \lambda, \quad (A - \lambda I)e_i = 0$ seems to give a reasonable way to construct improved subspace updates. May B_{λ_i} be an (easy to compute) approximation to $(A - \lambda_i I)^{-1}$. Add the approximate inverses applied to the residue to the search space: $\mathbf{V^{(k+1)}} = span(\mathbf{V^{(k)}}, B_{\lambda_1} r_1, \ldots, B_{\lambda_m} r_m)$, where $r_i := (A - x_i^T A x_i \; I)x_i$ with x_i the best eigenvector approximations available. An alternative is using $B_{\lambda_1} x_i$ directly, thus approximating 'shift and invert', but this has obvious stability problems. Until recently, the only method using approximate inverses was Davidson's method which simply uses the diagonal entries of A to compute B, and it was useful only for matrices from theoretical chemistry. While convergence was demonstrated to be rather fast, no analysis was available, and attempts to improve it by using better approximate inverses failed. In hindsight, the reason for this is quite clear, and some idea was there right from the start. If B is exact, then $B_{\lambda_i} r_i = x_i$, obviously not a good choice. So B must not be too good. On the other hand, if B is a poor approximation, this is not much better than taking r_i itself, which is the Krylow subspace calculation without the computational shortcuts. The annoying problem that improving B might reduce convergence was understood and overcome in [12], where it was proved that the exact way to define a correction equation is to project the correction problem into the space orthogonal to e_i, and e_i not being available, the space orthogonal to x_i will do fine, too. This leads to the improved definition of q_i:

$$[(I - x_i\, {x_i}^T)\; (\bar{A} - \bar{\lambda}_j^{(k)} I)\; (I - x_i\, {x_i}^T)]\; q_i = r_i$$

The projection $(I - x_i\, {x_i}^T)$ is not easy to incorporate into the matrix, but there is no need to do so. Because of $r_i \perp -x_i$,

$$q_i = (\epsilon + \bar{\lambda}_i) B^{-1}\, \vec{x}_i - B^{-1}\, A\, x_i \qquad \text{with} \qquad \epsilon = \frac{{x_i}^T\, B^{-1}\, r_i}{{x_i}^T\, B^{-1}\, x_i}$$

gives the proper solution of the projected equation.

There is no need to use the same type of approximate inverses throughout the computation. In some finite element test cases, the best efficiency has been achieved by starting out with a rather crude and simple choice for B (diagonal only) and getting more accurate as the eigenvector approximations improve [3]. This leaves the field of tuning the algorithm wide open.

Approximate inverses are, strictly speaking, not part of the eigenvalue algorithm but only a plug-in, but of course of highest importance for the efficiency. Therefore a few words on the topic seem appropriate. To build a good method, the literature on approximate inverses should be consulted.

In most cases, an approximate inverse will be constructed by extracting from A a structurally simpler (e.g. narrow banded, very sparse, ...) matrix that contains most of the information of A, and invert this accurately or again approximately (incomplete decomposition). With most matrices from quantum chemistry, taking for $\bar{A}$ a (possibly tapered) band with $k(1) \ll n\,, k(i+1) \leq k(i)$

$\bar{a_{ij}} = 0 \quad \text{for} \quad |i-j| > k(i), \qquad \bar{a_{ij}} = a_{ij} \quad \text{for} \quad |i-j| \leq k(i)$

gives a useful approximate inverse. The Davidson method is a special example, and often near optimal.
If A is not a stored matrix but only given as a procedure to calculate Ax from x, it rarely pays to extract $\bar{A}$. Here the approximate inverse can be calculated using a conjugate gradient method for solving an equation $(A - \lambda I)q = r$, and stopped at appropriate accuracy. This approach is widely used for eigenvalue problems from PDE's.

4 Problems of implementation and parallelization

The eigenvalue computation can be separated into actions in the n-dimensional space and those in the projected space. The former are generally simple, but time consuming, while the latter may be very complicated indeed, but take up little time. The n-dimensional operations consist of calculation of Ax, the solution of $(A - \lambda I)q = r$, calculation of scalar products and linear combinations of vectors. Except for very peculiar data structure of A, all this is done best by using the existing efficient implementations of linear algebra, BLAS and LAPACK for the sequential and PBLAS and ScaLAPACK for parallel computing. The ARPACK and PARPACK packages [1,8] give careful implementations of the Lanczos and Arnoldi method and may be used either as is or serve as a guideline and provide building blocks for other implementations.
There are some not so obvious details that need special attention. In concept, an orthonormal basis $\vec{w}_1^{(k)}, \ldots, \vec{w}_m^{(k)}$ of the subspace $\mathbf{V^{(k)}}$ is required to build the matrix $S^{(k)} := (W^{(k)})^T A W^{(k)}$, and this is usually done by applying the modified Gram-Schmidt method to a basis of $\vec{v}_1^{(k)}, \ldots, \vec{v}_m^{(k)}$ of $\mathbf{V^{(k)}}$. This is not necessary. May $V := [v_1, \ldots, v_m]$, T the eigenvectors of $\hat{A}$, $F = V^T V$, C the Cholesky decomposition of F and $R = C^{-1}$, then $W = V R$, and $S := R^T ((AV)^T V) R$. The calculation of the Ritz vectors can be done via $W^{(k)} T^{(k)} := V(RT)$, so that there is no need to actually compute the $\vec{w}_i^{(k)}$. All that is needed in the n-dimensional space is $V^T V$ and $V^T A V$, all other calculations being only in the low-dimensional subspace.
This saves about half the computations of the Gram-Schmidt method, but there is a problem. The Ritz projection is rather sensitive to errors in orthogonality in W, so the basis $\vec{v}_1^{(k)}, \ldots, \vec{v}_m^{(k)}$ must not be near degenerate. Otherwise, the culprit vector has to be removed or replaced by a new, truly independent direction. This can be organized by using an incremental Cholesky decomposition. The method is computationally similar to the original Gram-Schmidt method, so the numerical stability is less than that of the modified Gram-Schmidt method. Especially if a restart is to be done, the computation of the vectors forming the basis of the reduced space is a numerically sensitive step. Here, actual orthogonalization may pay.
The only computations in low dimensional space that may contribute to the computational load are the solution of the eigenvalue problem of S (e.g. by using some LAPACK procedure) and possibly the process to choose the approximations that will be put to further use.

All the n-dimensional linear algebra calculations can be distributed with benefit over different processors of parallel machines. As they have predictable computational effort, static load balancing will do. There are full codes, building blocks and development tools available for almost any architecture, but writing efficient parallel programs still requires skill and insight. The easy-to-use methods like HPF or virtual shared memory are considerably less efficient than explicit message passing, which is not an easy-to-use method. The speedup available depends on problem size, but there are examples of a speedup of 500 on a 512 processor machine.

5 Conclusions

The calculation of eigenvalues and -vectors of a large matrix is an old topic of mathematics, but there is still progress. While matrix transformation, direct iteration and the use of exact inverses have long been understood and are cast into up-to-date implementations, the proper use of approximate inverses is quite recent, and the implementations are less mature and not generally available. The building blocks for state-of-the-art code are there, but it still has to be put together. Parallel implementations may well exploit parallelism for the n-dimensional operations, while the parallelism for the low dimensional computations is tough to use at all.

References

1. R.B. Lehoucq, D.C. Sorensen, C. Yang, ARPACK Users' Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods. *http://www.netlib.org/sclapack/arpack_ug.ps.gz*
2. A. Basermann and B. Steffen, New Preconditioned Solvers for Large Sparse Eigenvalue Problems on Massively Parallel Computers, *Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing,* CD-ROM, SIAM , Philadelphia, (1997).
3. A. Basermann and B. Steffen, Preconditioned Solvers for Large Eigenvalue Problems on Massively Parallel Computers and Workstation Clusters, *Parallel Computing: Fundamentals, Applications and New Directions*, eds. E.H. D'Hollander, G.R. Joubert, F.J. Peters, and U. Trottenberg, Elsevier Science B. V., 565 (1998).
4. J.K. Cullum and R.A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*, Volume I: Theory, Birkhäuser, Boston Basel Stuttgart, 1985.
5. E.R. Davidson, The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real symmetric matrices, *J. Comp. Phys.* **17**, 87-94 (1975).
6. R.W. Freund and N.M. Nachtigal, QMR, A quasi minimal residual method for non-hermitian linear systems, Numer. Math. **60**, 315 (1991).
7. N. Kosugi, Modification of the Liu-Davidson method for obtaining one or simultaniously several eigensolutions of a large real symmetric matrix, *Comput. Phys.* **55**, 426-436 (1984).

8. K.J. Maschhoff, D.C. Sorensen, A Portable Implementation of ARPACK for Distributed Memory Parallel Architectures, *http://www.caam.rice.edu/ kistyn/parpack_home.html*
9. B.N. Parlett *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, PA (1998) Updated reprint of the 1980 Prentice Hall edition .
10. H. Rutishauser, Computational aspects of F. L. Bauer's simultaneous iteration method, Numer. Math. **13**, 4-13 (1969).
11. B. Steffen, *An Improved Version of the Semi-analytical Eigenvector Processor* KFA-ZAM-IB-9105, (1991).
12. G.L.G. Sleijpen and H.A. van der Vorst, A Jacobi-Davidson iteration for linear eigenvalue problems, SIAM J. Matrix Anal. Appl. **17**, 401 (1996).
13. J. Tückmantel, *An Improved Version of the Semi-analytical Eigenvector Processor SAP*, CERN/EF/RF 85-4, (1985).